**GitOps Workflow using ArgoCD on Kubernetes**

**Loom video:** https://www.loom.com/share/f2a737a2a30d4e5c9c54bb504842f11b?sid=16e7217d-d104-457d-bd14-bd112eb3a298

**Abstract:**

This project demonstrates the implementation of GitOps practices using ArgoCD on a local Kubernetes cluster (Minikube). The goal was to automate Kubernetes deployments by storing application manifests in GitHub and enabling continuous synchronization with ArgoCD. This ensures reliable, consistent, and automated deployments for modern cloud-native applications.

**Introduction:**

GitOps is an operational model for Kubernetes where Git acts as the single source of truth. ArgoCD, a declarative GitOps tool, continuously monitors Git repositories and keeps Kubernetes clusters synchronized. This project highlights how GitOps simplifies deployment, improves automation, and provides version control for infrastructure.

**Tools Used:**

* Docker (container runtime for Minikube)
* Minikube (local Kubernetes cluster)
* kubectl (Kubernetes CLI)
* Git & GitHub (version control & repo for manifests)
* ArgoCD (GitOps continuous delivery tool)

**Steps Involved in Building the Project:**

1. Installed prerequisites: Docker, Minikube, kubectl, and Git.
2. Started a Kubernetes cluster with Minikube and verified node status.
3. Installed ArgoCD in the 'argocd' namespace and verified pods and services.
4. Exposed ArgoCD UI using kubectl port-forward and logged in with admin credentials.
5. Created Kubernetes manifests (Deployment & Service) and pushed them to GitHub.
6. Connected GitHub repo to ArgoCD and enabled Auto-Sync.
7. Tested GitOps by scaling Nginx replicas from 1 → 2 via Git commit, observed ArgoCD auto-sync update the cluster.

**Conclusion:**

The project successfully implemented GitOps using ArgoCD and Kubernetes. By storing application manifests in GitHub and enabling automated synchronization, we achieved continuous delivery, version control, and simplified deployment management. This demonstrates how GitOps enhances reliability and consistency in Kubernetes environments





